

Soit un système avec des processus (threads) et des ressources à un instant donné S avec 4 structures de données :

E : nombre de ressources maximum disponible (ne change pas durant l'exécution) :

R1	R2	R3	R4	R5

A : nombre de ressources disponibles à l'instant S de l'exécution :

R1	R2	R3	R4	R5
0	0	x	1	1

Alloc : tableau de l'allocation de chaque processus en fonction des ressources disponibles à l'instant S de l'exécution:

	R1	R2	R3	R4	R5
P1	1	0	2	1	1
P2	2	0	1	1	0
P3	1	1	0	1	0
P4	1	1	1	1	0

Req : tableau des ressources qui seront demandées par chaque processus dans le futur i.e. après S:

	R1	R2	R3	R4	R5
P1	0	2	0	0	1
P2	0	2	1	0	0
P3	1	0	3	0	0
P4	0	0	1	1	1

Soit x=1, S est-il sur ?

Rép. On va appliquer l'algorithme du banquier :

Itération no 1 :

- Est-ce que l'ordonnanceur peut allouer à aux moins un des 4 processus les ressources demandés après l'instant S ?

A :	R1	R2	R3	R4	R5	
	0	0	1	1	1	
Req(P1) :	0	2	0	0	1	non
Req(P2) :	0	2	1	0	0	non
Req(P3) :	1	0	3	0	0	non
Req(P4) :	0	0	1	1	1	oui

On voit que les ressources demandées par P4 sont suffisantes, donc **oui**

- Mise à jour de A quand P4 aurait terminé avant d'aller à la prochaine itération :

À la fin de son exécution, P4 va donc libérer les ressources Alloc(P4) et celle acquises dans A via Req(P4).

À la fin on peut dire que la nouvelle valeur de A nommons la A' sera égale à $A + \text{Alloc}(P4)$ donc :

Alloc(P4) :	1	1	1	1	0
A :	0	0	1	1	1
A' :	1	1	2	2	1

Itération no 2 :

- D'abord A' devient A pour cette nouvelle itération et on se concentre maintenant sur P1, P2 et P3.
- Est-ce que l'ordonnanceur peut allouer à P1 les ressources demandés après l'instant S ?

	R1	R2	R3	R4	R5	
A :	1	1	2	2	1	
Req(P1) :	0	2	0	0	1	non
Req(P2) :	0	2	1	0	0	non
Req(P3) :	1	0	3	0	0	non

S est non sur !

Maintenant pour x= 2

Itération no 1 :

- Est-ce que l'ordonnanceur peut allouer à au moins un des 4 processus les ressources demandés après l'instant S ?

	R1	R2	R3	R4	R5	
A :	0	0	2	1	1	
Req(P1) :	0	2	0	0	1	non
Req(P2) :	0	2	1	0	0	non
Req(P3) :	1	0	3	0	0	non
Req(P4) :	0	0	1	1	1	oui

On voit que les ressources demandées par P4 sont suffisantes, donc **oui**

- Mise à jour de A quand P4 aurait terminé avant d'aller à la prochaine itération :

À la fin de son exécution, P4 va donc libérer les ressources Alloc(P4) et celle acquises dans A via Req(P4).
 À la fin on peut dire que la nouvelle valeur de A nommons la A' sera égale à A + Alloc(P4) donc :

Alloc(P4) :	1	1	1	1	0
A :	0	0	2	1	1
A' :	1	1	3	2	1

Itération no 2 :

- D'abord A' devient A pour cette nouvelle itération et on se concentre maintenant sur P1, P2 et P3.
- Est-ce que l'ordonnanceur peut allouer à P1 les ressources demandés après l'instant S ?

	R1	R2	R3	R4	R5	
A :	1	1	3	2	1	
Req(P1) :	0	2	0	0	1	non
Req(P2) :	0	2	1	0	0	non
Req(P3) :	1	0	3	0	0	oui

On voit que les ressources demandées par P3 sont suffisantes, donc **oui**

- Mise à jour de A quand P3 aurait terminé avant d'aller à la prochaine itération :

À la fin de son exécution, P3 va donc libérer les ressources Alloc(P3) et celle acquises dans A via Req(P3).
 À la fin on peut dire que la nouvelle valeur de A nommons la A' sera égale à A + Alloc(P3) donc :

Alloc(P3) :	1	1	0	1	0
A :	1	1	3	2	1
A' :	2	2	3	3	1

Itération no 3 :

- D'abord A' devient A pour cette nouvelle itération et on se concentre maintenant sur P1 et P2.
- Est-ce que l'ordonnanceur peut allouer à P1 les ressources demandés après l'instant S ?

	R1	R2	R3	R4	R5	
A :	2	2	3	3	1	
Req(P1) :	0	2	0	0	1	oui
Req(P2) :	0	2	1	0	0	oui

On voit que les ressources demandées par P1 ou P2 sont suffisantes, donc **on va choisir P2**

- Mise à jour de A quand P2 aurait terminé avant d'aller à la prochaine itération :

À la fin de son exécution, P2 va donc libérer les ressources Alloc(P2) et celle acquises dans A via Req(P2).
 À la fin on peut dire que la nouvelle valeur de A nommons la A' sera égale à A + Alloc(P2) donc :

Alloc(P2) :	2	0	1	1	0
A :	2	2	3	3	1
A' :	4	2	4	4	1

Itération no 4 :

- D'abord A' devient A pour cette nouvelle itération et on se concentre maintenant sur P1.
- Est-ce que l'ordonnanceur peut allouer à P1 les ressources demandés après l'instant S ?

	R1	R2	R3	R4	R5	
A :	4	2	4	4	1	
Req(P1) :	0	2	0	0	1	Évidemment Toujours Oui

On voit que les ressources demandées par P1 sont suffisantes.

- Mise à jour de A quand P1 aurait terminé avant d'aller à la prochaine itération :

À la fin de son exécution, P1 va donc libérer les ressources Alloc(P1) et celle acquises dans A via Req(P1).
 À la fin on peut dire que la nouvelle valeur de A nommons la A' sera égale à A + Alloc(P1) donc :

Alloc(P1) :	2	0	1	1	0
A :	4	2	4	4	1
A' :	6	2	5	5	1

S est non sur ! Et évidemment pour $x > 2$!